# Mybatis 3.0
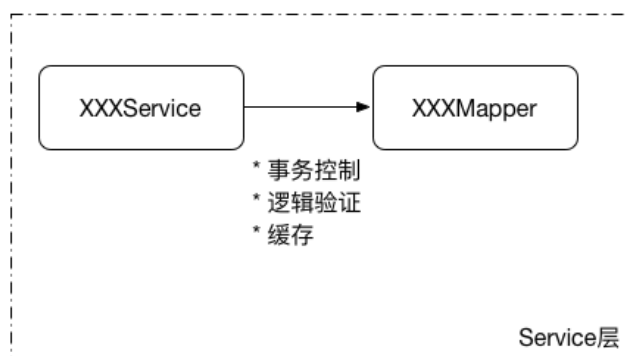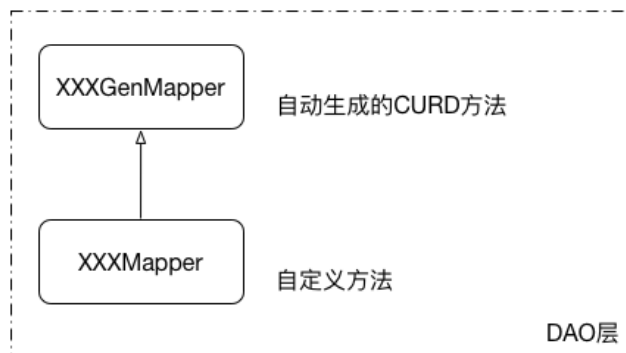
by Shiy 2017.4

## 一、 Mybatis 3.0 变化

- DAO -> Mapper 纯接口化
- SQL配置xml与Mapper绑定
  - namespace与类名绑定
  - id与方法绑定
- SQL配置文件语法
- 零配置的annotation方式

## 二、 使用方法

### 0. 整体





uml

### 1. 自动生成插件

- pom.xml 配置插件

```xml
<plugin>
    <groupId>org.mybatis.generator</groupId>
    <artifactId>mybatis-generator-maven-plugin</artifactId>
    <version>1.3.5</version>
    <configuration>
        <configurationFile>${basedir}/gen/generatorConfig.xml</configurationFile>
        <overwrite>true</overwrite>
    </configuration>
    <dependencies>
        <dependency>
            <groupId>mysql</groupId>
```

```xml
            <artifactId>mysql-connector-java</artifactId>
            <version>5.1.41</version>
        </dependency>
        <dependency>
            <groupId>cn.shiy.mybatis</groupId>
            <artifactId>generator-plugin</artifactId>
            <version>1.0</version>
        </dependency>
    </dependencies>
</plugin>
```

- 生成命令

```
mvn mybatis-generator:generate
```

## 2. generatorConfig配置

```xml
<table schema="" tableName="customer" domainObjectName="Customer">
    <property name="useActualColumnNames" value="true"/>
    <generatedKey column="ID" sqlStatement="MySql" identity="true"/>
    <columnOverride column="type" javaType="int" jdbcType="TINYINT"/>
    <!-- DATETIME类型使用TIMESTAMP即可 -->
    <columnOverride column="createtime" javaType="Date" jdbcType="TIMESTAMP"/>
    <ignoreColumn column="tobeignore"/>
</table>
```

## 3. 完整示例

1. 数据库表结构

### customer

| | Field | Type | Allow Null | Default Value |
|---|---|---|---|---|
| P | id | bigint(20) | No | |
| | name | varchar(255) | Yes | |
| | type | tinyint(4) | Yes | |
| | createtime | datetime | Yes | |
| | tobeignore | varchar(255) | Yes | |

table

2. 上述generatorConfig配置

3. 执行生成命令

4. 得到的代码在 `target/generated-sources/mybatis-generator`

   - sqlmap/XXXMapper.xml

   - {package}/dao

   - {package}/entity

5. 复制代码到src

## 4. 使用注意

- Annotation方式(仅限简单的语句)

```java
/**
 * mapper方式的UserDAO
 * @author shiy
```

```java
 */
public interface UserMapperDAO {


    @Select("select * from user where id= #{id}")
    User getUser(long id);

    @Insert("insert into user(name,age,birthday,gender) values(#{name},#{age},#{birthday},#{gender})")
    @SelectKey(statement = "SELECT LAST_INSERT_ID()",before=false,resultType=Long.class,keyProperty = "id")
    long insertUser(User user);
}
```

- 获取上次搜索的总数据条数(无视limit)

```java
CustomerMapper mapper = Container.getBean(CustomerMapper.class);
CustomerExample ex = new CustomerExample();
ex.setNeedFoundRows(true);
ex.setLimitStart(0);
ex.setLimitEnd(1);
ex.or().andIdBetween(13L, 15L);
List<Customer> list = mapper.selectByExample(ex);
assertEquals(mapper.countLastSelect(), 3);
```

- 自定义SQL

  - 新建UD_XXXMapper.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="cn.shiy.test.mybatis.dao.CustomerMapper">

  <select id="getCountOfCustomer" resultType="java.lang.Long">
    select count(*) from customer
  </select>

</mapper>
```

  - Mapper添加自定义方法

```java
public interface CustomerMapper extends CustomerGenMapper {

    long getCountOfCustomer();
}
```